# Ubiquitous Service Platform using Contextual Information

**Hyung-Min Yoon[a], Seong-Hun Jeong[b], Oh-Young Kwon[c], and Tack-Don Han[d]**

[abd] *Dept. of Computer Science, Yonsei University*
*134 Shinchon-dong, Seodaemoon-gu, Seoul, 120-749, South Korea*
*Tel: +82-2-2123-3524, Fax: +82-2-365-2579, E-mail: {yoonhm[a], shjeong[b], hantack[d]}@kurene.yonsei.ac.kr*

[c] *Dept. of Computer Engineering, Korea University of Technology and Education*
*Gajeonri 307, Byungcheon-Myeon, Cheonan City, Choongchungnam-Do, 330-708, South Korea*
*Tel: +82-41-560-1354, Fax: +82-41-564-3261, E-mail: oykwon@kut.ac.kr*

**Abstract**

*Network infrastructure has spread to an unprecedented extent and is used in various computing devices, such as smart appliances, smart phones, and embedded devices with sensors, which have all been appearing in the computing environment. To accommodate this trend, for a more intelligent service environment, the service platform needs to have abilities that facilitate the operation between services, dynamically share distributed computing resources, and manage appropriate contextual information. We have simulated a service platform to provide intelligent services using contextual information after having implemented the context management service. The context management service gathers raw contextual information from sensors and stores these in the context database. For a consistent basis of contextual information, time and location are used as the key values of the contextual information. The context management service also performs normalization to provide computable contextual information to context-aware applications.*

*In this paper, a service platform based on Jini technology is proposed for constructing an interoperable, dynamic, and intelligent service environment using contextual information.*

***Keywords:***

Ubiquitous computing, intelligent service, contextual information, Jini, context-aware applications, ubiquitous service platform

## Introduction

Through the proliferation of the Internet and mobile networks and the combination of wireless networks, computing environments are becoming more ubiquitous. New devices and services of many different types are appearing, and these changes show a tendency to be smaller, wearable, and intelligent. In particular, services are developing in the direction of providing intelligent services related to contextual information. This technology trend presents a computing environment that provides various services to users intelligently and seamlessly, regardless of the location and change of environment. Satisfying the requirements of new computing environments requires that they have the capabilities for code mobility and automatic service discovery [10]. For these capabilities, Jini technology [1], as a middleware framework, has the appropriate network structure for ubiquitous computing and provides mobility of application and dynamic resource sharing [12]. As well as Jini technology, various middleware structures [6], such as Universal Plug and Play (UPnP) [7] and Salutation [8] have been proposed and used with service discovery mechanisms.

In these environments, contextual information gathered from various sensors is used to provide intelligent services that change with the user's environment [13]. This information is applicable to efficient and convenient computing environments. The contextual information acquired by wearable computers is user's bio-information, user activities, locations, directions, weather, and surrounding information. This information demands a policy regarding gathering, storing, classification, transformation, aggregation, normalization, and standardization. In particular, normalization and standardization are important to provide a consistent basis of contextual information and persistence of computable information. Contextual information is classified into numerical values and non-numerical values. Non-numerical values are required to be transformed into quantitative values [11]. Because various different sensors acquire it, the contextual information is of various different types, and is not standardized [11]. For efficient use of this contextual information, the system platform must have the ability to analyze the information [11]. In order to implement this function, many efforts have been attempted to gather and provide contextual information while using agent [14]. In this paper, contextual information is classified and sorted by time and location. To use the contextual information, a context management service is designed and a dynamic service platform [9] based on Jini technology is proposed.

## Contextual Information Analysis

### Classification of contextual information

Contextual information can not only be gathered from the same sensors embedded or equipped in the system but also from various sensors embedded in mobile environments. Contextual information is divided into human context, environment context, system information, and digital information as shown in Table 1.

*Table 1 - Four types of contextual information*

| Type | Group | Examples |
|---|---|---|
| Human Context | Human Relation | Digital Relation: Human–Digital Information<br>Physical Relation: Human–Human Device<br>Human Relation: Human–Human<br>Social Relation: Human–Society |
| | Human Activity | Moving velocity<br>Activity: Working, Cooking, Shopping, Moving, ... |
| | Human Sense | Pleasure, Sorrow, Anger |
| | Human Bio | Body Information: Weight, Height, Form, ...<br>Health Information: Exercise, Caloric intake |
| Environment Context | Nature | Weather, Seasons, Sunlight, Climate, Noise, Temperature |
| | Physical Object | Furniture, Car, Book, ... |
| | Human Generating | Traffic condition, Crowd, Confusion, History, ... |
| System Information | Service | Service state, Service attribute, History of use |
| | Network | Network connectivity, Communication cost, Communication bandwidth |
| | Device Information | Battery, Display, Audio |
| Digital Information | Web Information | URL, U-ID |
| | Document Information | Author, Date of creation, File location |

This requires definite criteria, that is, contextual information, to use in the computing system. Time and location as the keys of the context values are suitable for classifying contextual information. Time is subdivided into period and moment. Location is subdivided into geographical location and digital location with two divisions of position and area as shown in Table 2.

*Table 2 - Criteria of contextual information*

| Criteria | Subdivision | |
|---|---|---|
| Time | A. period | a(duration), b(year, month, day, hour), c(season) |
| | B. moment | a(specific time) |
| Location | A. position | a(TM, UTM, ...) |
| | B. area | a(address, domain), b(area presented by coordinates) |
| | C. digital position | a(URL), b(directory and file name) |
| | D. digital area | a(domain name) |

### Calculation of contextual information

Numerical contextual information can be applied to operations such as "+, -, *, /". Moreover, these can be analyzed and applied to many forms of mathematical models. In addition, there exists non-numerical contextual information. It is important to use appropriate classification and interpretation policies for contextual information in the computing system [11].

Numerical contextual information consists of values that are not necessary for additional work or values that are required to perform standardization operations.

*Table 3 - Numerical and non-numerical contextual information*

| Type | Examples | Characteristics |
|---|---|---|
| Numerical contextual information | The number of men in a specific area | No need of standardization |
| | Location (coordinates, address, ...)<br>Moving velocity (km/s, miles/s, ...)<br>Direction (NSEW, Degree, ...) | Need of standardization |
| Non-numerical contextual information | Activity (working, moving, playing, sleeping, eating) | Need of evaluation and standardization |

Some numerical contextual information may be of different types according to nationality, region or culture. Non-numerical contextual information can be transformed to calculable values by simple operations that assign priorities or logical operations that result in true/false values.

## Ubiquitous Service Platform

### Jini framework

Jini, a service framework based on Java technology, has enabled the development of highly distributed and dynamic service environments. Discovery, join, service lookup, and service registration are the main functions of Jini [3] for remote service invocation [4]. Services in Jini networks can dynamically interact with each other, without additional connection effort, and Jini supports dynamic interaction as well as plug-and-play [1]. In addition, Jini applies leasing technology and defines distributed event and transaction mechanisms [3]. Due to these characteristics of Jini, services can be interoperable and reusable.

### Ubiquitous Service Platform structure

The Ubiquitous Service Platform consists of three components. One is the client browser to use the services. Second is the embedded middleware as the service provider. The key component is the lookup service, which, as service manager, connects the service provider and client [3].

#### *Client Browser*

The client browser operates the setup of the discovery protocol and performs any necessary service lookup [2]. These services are initiated when the client browser executes the service proxy downloaded from the lookup service [2]. The service proxy can notify the lookup service of the event and receive the remote event from the lookup service via the client browser [2].

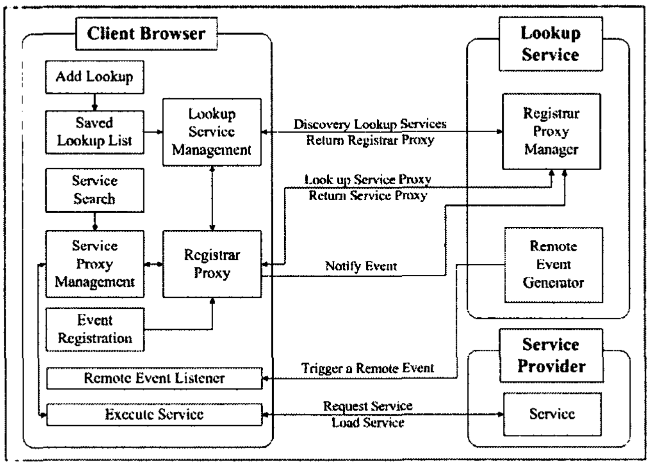


*Figure 1 - Client Browser Structure*

#### *Lookup Service*

Our lookup service is based on "reggie" of Sun Microsystems Inc. We added a Security Manager and a Device Manager to "reggie" to simulate a lookup service. Using encryption, the Security Manager performs authentication incorporating ID and password into the discovery protocol. The Device Manager manages the list of connected devices as service provider to search for devices.

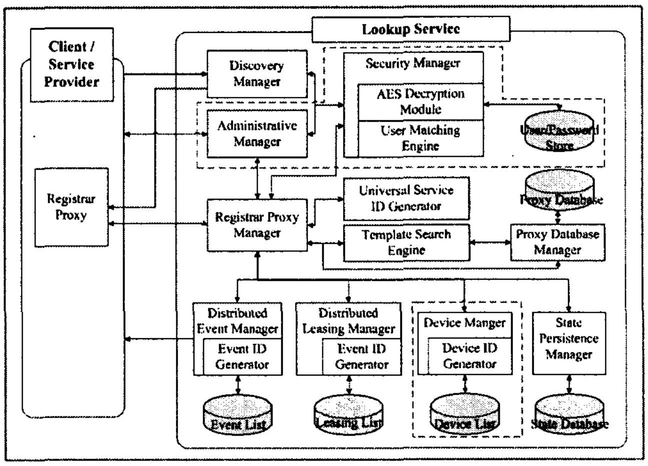


*Figure 2 - Lookup Service Structure*

#### *Embedded Middleware*

Embedded middleware acts as a service provider to enable devices to interact and provide remote services [9]. Services are able to call system libraries developed in Java, C and C++ with JNI technology [5]. The assignment of remote commands to devices on networks, and plug-and-play are available to devices with embedded middleware using Jini Technology. It provides a "Context Management Service" to context-sensitive services for dynamic and intelligent service.

### Context Management Service

Each device stores the contextual information that was gathered from their sensors in an independent "Context DB." We implemented the "Context Management Service," which collects the raw contextual information and transforms it into the appropriate form to be used by the applications. The "Context Management Service" is composed of a "Context Gathering Module," a "Context Transformation Module," and a "Context DB."

The "Context Gathering Module" requests the devices to sense the raw contextual information from the sensors, and then it collects the raw contextual information. Furthermore, it can receive an event notification of context change from each device using the "Context Management Service." It classifies the raw contextual information by time and location, and transfers the information to the "Context Transformation Module."

The "Context Transformation Module" transforms the classified raw contextual information into computable information using a "Regulation" and "Quantification" process. The "Regulation" process transforms the numerical values to the common types used for the same purpose in the various systems. The "Quantification" process transforms non-numerical values to quantitative values or logical values. These two processes enable the analysis of the contextual information. The "Context Transformation Module" combines the computable raw contextual information, environmental and system context, and then uses mathematical and logical operations to produce the normal contextual information.

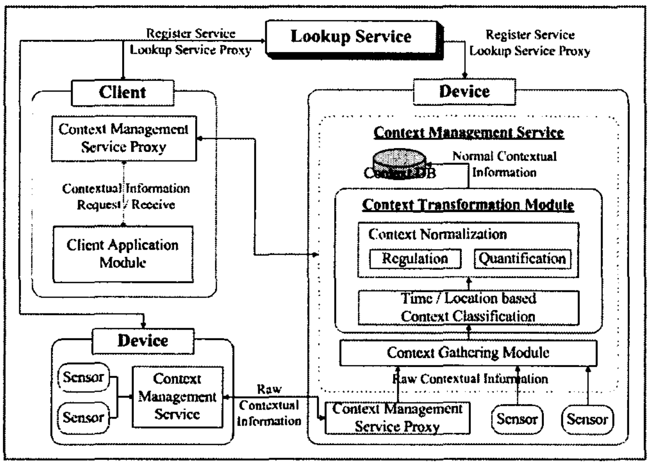


*Figure 3 - Context Management Service System Structure*

The normal contextual information is stored in the "Context DB" from where it is used by the applications. A contextual information transformation example is shown in Table 4.

*Table 4 - Contextual Information Transformation Example in Personal Digital Assistant (PDA)*

| Time & Location | Device | Raw Contextual Information / Regulation Quantification | Normal Contextual Information |
|---|---|---|---|
| 2003/09/20 20:00 KST Rm. 202, ASIC Lab., Yonsei Univ. | Air-conditioner | Temperature – 70°F, Latest connected users – 5 people | Users in the service area – 5 people, Temperature in the room – 21℃, User activity – Studying |
| | | Temperature – 21℃ | |
| | GPS | Position – 126°34'35.12"E. Long., 37°34'45.54"N. Lat. | |
| | | None | |
| | PDA | Running Program – Word Processor | |
| | | Running Program Category Code : 01 | |
| 2003/09/20 20:01 KST Rm. 202, ASIC Lab., Yonsei Univ. | GPS | Position – 126°34'35.12"E. Long., 37°34'45.56"N. Lat. | Velocity (inferred from position) – 1.5 m/s |
| | | None | |

The "Context Management Service" is currently a prototype and performs the transformation shown in Table 4.

## Simulation Environment

The lookup service and the embedded middleware were located in three different devices to simulate the Ubiquitous Service Platform. The client browser was installed in a mobile device with a wireless network. First, on moving, the mobile device searches neighboring Context Management Services. Then, the client browser in the mobile device receives the change of contextual information and retrieves the transformed contextual information. The client browser stores the contextual information for future use by its services.

## Discussion

In this paper, a Ubiquitous Service Platform is proposed to manage the distributed contextual information dynamically to provide interactive and intelligent services to users. The Ubiquitous Service Platform consists of the client browser, which is able to provide service to users, embedded middleware, which gathers the contextual information from various devices, and the lookup service, which connects these. To manage all forms of contextual information we first classified the contextual information by time and location, and then the contextual information was transformed to calculable values by regulation and quantification. The processes are performed in the Context Management Service so that raw contextual information is normalized for use as in a standard information form.

Contextual information increases as more embedded devices are added, but services will be more intelligent using this information. We intend to improve the Ubiquitous Service Platform to be suitable for future computing environments by defining the management mechanism of all the forms of contextual information.

## References

[1] J. Waldo (1999). "The Jini Architecture for network-centric computing", *Communications of the ACM*, Vol. 42, pp. 76-82.

[2] Sing Li (2000). *Professional Jini.* Wrox Press.

[3] K. Arnold, B. O'Sullivan, R.W. Scheifler, J. Waldo, and A. Wollrath (1999). *The Jini™ Specification.* Reading, MA: Addison-Wesley.

[4] Sun Microsystems, Inc. (2002). "Java Remote Method Invocation Specification," Revision 1.8, available online: http://java.sun.com/j2se/1.4/docs/guide/rmi/index.html

[5] Beth Stearns (2000). "Java Native Interface," available online:

http://java.sun.com/docs/books/tutorial/native1.1/index.html

[ ] E. Guttman (1999). "Service Location Protocol: Automatic Discovery of IP Network Services," *IEEE Internet Computing*, Vol. 3, pp. 71–80.

[ ] Microsoft Corporation (2000), "Universal Plug and Play Device Architecture," Version 1.0, available online: http://www.upnp.org/download/UPnPDA10_20000613.htm.

[ ] E. Guttman, C. Perkins, J. Veizades, and M. Day (1999). "Service Location Protocol," Version 2, Internet Engineering Task Force (IETF), RFC 2608.

[ ] Christian Becker, Gregor Schiele, Holger Gubbels, and Kurt Rothermel (2003). "BASE - A Micro-broker-based Middleware For Pervasive Computing," *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom'03),* Fort Worth, Texas, pp. 443-451.

[ C]T. Kindberg, and A. Fox (2002). "System Software for Ubiquitous Computing," *IEEE Pervasive Computing,* Vol. 1, pp. 70-81.

[11]Jani Mantyjarvi, Johan Himberg, and Pertti Huuskonen (2003). "Collaborative Context Recognition for Handheld Devices," *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom'03),* Fort Worth, Texas, pp. 161-168.

[12]Rahul Gupta, Sumeet Talwar, and Dharma P. Agrawal (2002). "Jini Home Networking: A Step toward Pervasive Computing," *IEEE Computer*, Vol. 35, pp.34-40.

[13]Patrick Brézillon (2003). "Focusing on Context in Human-Centered Computing," *IEEE Intelligent Systems*, Vol. 18, pp. 62-66.

[14]Anand Ranganathan, and Roy H. Campbell (2003). "A Middleware for Context-Aware Agents in Ubiquitous Computing Environments," *Proceedings of the ACM/IFIP/USENIX International Middleware Conference,* Rio de Janeiro, Brazil, pp. 143-161.